PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-063576
(43)Date of publication of application : 06.03.1998

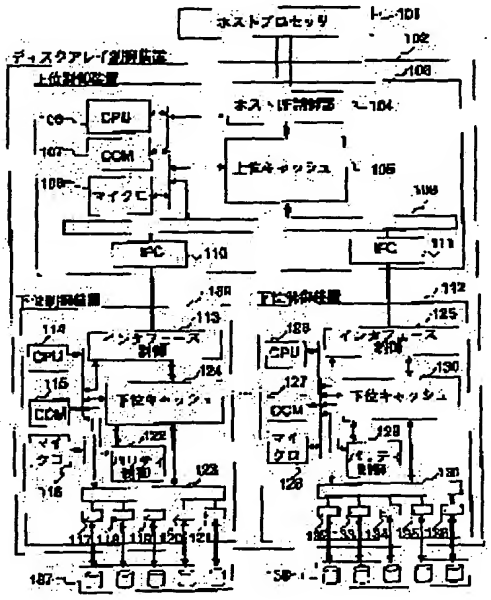
(51)Int.Cl. G06F 12/08
G06F 12/08
G06F 3/06
G06F 3/06
G06F 12/16

(21)Application number : 08-224792
(22)Date of filing : 27.08.1996

(71)Applicant : HITACHI LTD
(72)Inventor : TAKAMOTO YOSHIFUMI

(54) HIERARCHICAL DISK DRIVE AND ITS CONTROL METHOD

(57)Abstract:
PROBLEM TO BE SOLVED: To improve the hit rate of caches by performing such control that data are not stored in a low-order cache when transferred to a high-order cache.
SOLUTION: The caches are made hierarchical between a host controller 103 and slave controllers 139 and 112 and have the constitution flexibly altered according to a request access time and cost. Here, caches 105, 124, and 130 are provided in the host controller 103 and slave controllers 139 and 112, and control is so performed that data are not stored in the caches repeatedly. Namely, the controllers which are made hierarchical are provided with memories for storing the hierarchy levels of their controllers and when data are read out, the hierarchy level in the memory is read out to performing control so that only the highest-order controller stores the data in its cache. Therefore, the repetitive storage of data in caches is suppressed, so that the caches can effectively be used.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（A）　(11)特許出願公開番号
特開平10－63576
(43)公開日　平成10年(1998)３月６日

(51)Int.Cl.[6]		識別記号	庁内整理番号	ＦＩ		技術表示箇所
Ｇ０６Ｆ	12/08		7623－5B	Ｇ０６Ｆ	12/08	Ｆ
		３２０	7623－5B			３２０
	3/06	３０２			3/06	３０２Ａ
		５４０				５４０
	12/16	３２０	7623－5B		12/16	３２０Ｄ

審査請求　未請求　請求項の数４　ＯＬ　（全 10 頁）

(21)出願番号　特願平8－224792

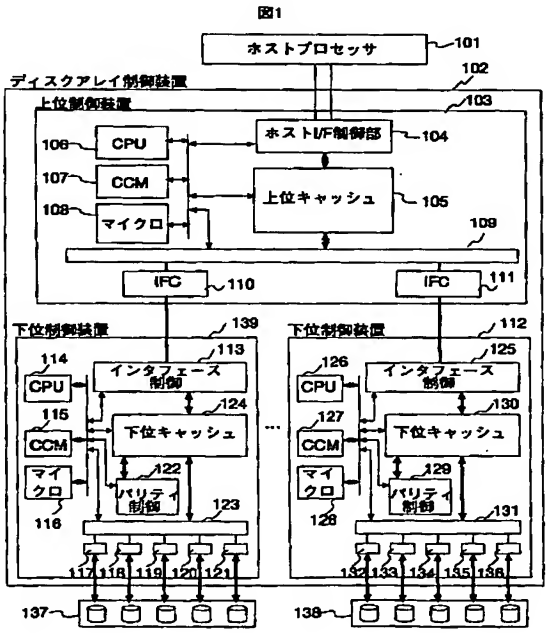
(22)出願日　平成８年(1996)８月27日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(72)発明者　高本　良史
東京都国分寺市東恋ケ窪一丁目280番地
株式会社日立製作所中央研究所内
(74)代理人　弁理士　小川　勝男

(54)【発明の名称】　階層ディスク装置およびその制御方法

(57)【要約】
【課題】　階層化されたディスクキャッシュにおいて、各キャッシュ間でデータの重複が発生しキャッシュの利用効率が低下するのを抑える。
【解決手段】　階層化された制御装置のそれぞれに、自制御装置の階層レベルを格納するメモリを設け、データの読み出し時に前記メモリ内の前記階層レベルを読み出し、最上位制御装置でなければ自キャッシュにはデータを格納しない制御を行う。
【効果】　各キャッシュ間で格納されているデータの重複が発生しないため、キャッシュの利用効率が向上する。

図1



【特許請求の範囲】

【請求項１】ホストプロセッサに接続された最上位制御装置と、前記最上位制御装置に接続された複数の下位制御装置と、前記複数の下位制御装置に接続された複数のディスク装置と、前記最上位制御装置および前記複数の下位制御装置に設けられたディスクキャッシュからなる階層ディスク装置であって、

前記下位制御装置は、前記ホストプロセッサからの読み込みデータを、ディスクキャッシュに格納すること無く前記上位制御装置へ転送する手段と、

前記最上位制御装置は、前記読み込みデータを、ディスクキャッシュに格納し、かつ、前記ホストプロセッサへ転送する手段を有する階層ディスク装置。

【請求項２】複数のディスク装置を並列に動作させるとともに、冗長データを格納するディスクアレイ制御装置であって、ホストプロセッサに接続された最上位制御装置と、前記最上位制御装置に接続された複数の下位制御装置と、前記複数の下位制御装置に接続された複数のディスク装置と、前記最上位制御装置および前記複数の下位制御装置に設けられたディスクキャッシュからなる階層ディスク装置において、

前記ホストプロセッサからのデータ読み込み時に前記階層レベルが最上位制御装置かどうかを、前記最上位制御装置および前記複数の下位制御装置が有する階層レベルを示す識別子から判断し、

最上位制御装置でなければ前記読み込みデータをディスクキャッシュに格納しない制御を行うことを特徴とする階層ディスクキャッシュ制御方法。

【請求項３】前記ホストプロセッサからデータの書き込み要求が発生した時、冗長データを生成する制御装置は、上位制御装置に対して最新のデータが存在するかどうか検索し、

存在すれば上位制御装置のディスクキャッシュからから下位制御装置のディスクキャッシュに前記データを転送することを特徴とする請求項２記載の階層ディスクキャッシュ制御方法。

【請求項４】前記最上位制御装置のキャッシュが障害を起こした場合、前記最上位制御装置に接続された複数の下位制御装置の前記識別子を、最上位制御装置を示す識別子に変更することを特徴とする請求項２記載の階層ディスクキャッシュ制御方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、ディスク制御装置に係り、階層化されたディスクキャッシュの利用効率を高め性能を向上させるの階層ディスク制御装置およびその制御方法に関する。

【０００２】

【従来の技術】一般的にコンピュータシステムは、プロセッサと２次記憶装置から構成されている。主として使用される２次記憶装置は磁気ディスク装置である。現在、ディスク記憶装置の容量の伸び率は極めて高いが、メカニカルな動作を伴う磁気ディスク装置の性能はプロセッサ性能の伸び率ほど高くない。その課題を解決する方法として、以下の２つの方法が知られている。

【０００３】第一の方法は、ディスク制御装置内にディスクキャッシュを設ける方法である。ディスクキャッシュは、一度読み込まれたデータを半導体メモリであるディスクキャッシュに格納しておき、同一のデータに対して再度読み込み要求が発生したときにはディスクキャッシュからデータを取り出すことで、ディスク装置へのアクセスを削減し高速化を行う方法である。キャッシュを増やすほどヒット率が向上するため性能も向上するが、容量当たりのコストが磁気ディスクに比べ高価であるため、通常は磁気ディスク容量の数１０分の１から数１００分の１の容量が設けられている。キャッシュ制御方法は、最も頻繁にアクセスされる順に格納しておき、最もアクセスされなかったデータの追い出しを行う。これをＬＲＵ処理といい、キャッシュ制御方法では一般的である。これらの方法に関して、特開平６－２３１０４５号公報に記載がある。

【０００４】第二の方法は、磁気ディスクを並列に制御することで性能を向上させる方式である。D.Patterson、G.Gibson、and R.H.Kartzらによる" A Case for RedundantArrays of Inexpensive Disks(RAID)、in ACM SIGMOD Conference、Chicago、IL"、 PP.109-116 (June 1988) (以下、第１の参考文献と呼ぶ) では、複数のディスクドライブにデータを分散して配置することでディスク内に格納されたデータへのアクセス時間を短縮し、かつパリティあるいはＥＣＣと呼ばれる冗長データを格納することで信頼性も高めるＲＡＩＤというディスクアレイの構成技術が紹介されている。

【０００５】つまり、アレイディスクでは、複数のディスクドライブに対して並列に入出力を行うので、データの読み出しあるいは書き込みは高速となり、また、ディスクドライブに障害が発生したときでもパリティと障害ディスクドライブ以外のデータから、障害ディスクドライブのデータを回復することができる。

【０００６】より具体的には、上記文献では、データの格納方法によりＲＡＩＤレベルを複数に分類している。そのうち、製品で多く使用されるＲＡＩＤレベルは、ＲＡＩＤ１、ＲＡＩＤ３、ＲＡＩＤ５である。ＲＡＩＤ１はミラーリングであり、２台のディスクに同一データを格納することでディスクの障害に対する信頼性を高めている。読み出し時には、２台のディスクの内どちらか早いほうのディスクから読むことで単一ディスクに比べ高速である。ＲＡＩＤ３は、単一データをビットあるいはバイト毎にストライピングし複数のディスクに並列に読み書きすることで、データ転送性能を向上可能である。ＲＡＩＤ５は入出力ブロックを単位として複数のディス

クにストライピングを行うレベルである。

【０００７】ＲＡＩＤ３では単一入出力要求を小さな単位でストライピングするのに対し、ＲＡＩＤ５はそれよりも大きな単位でストライピングを行う。ＲＡＩＤ３は単一ユーザの大規模データ入出力を高速化することが主な目的であるが、ＲＡＩＤ５は多数ユーザの小規模データ入出力を並列に実行することが主な目的である。従って、ＲＡＩＤ３は特に大規模なデータ入出力が要求されるマルチメディアや科学技術計算に用いられる。ＲＡＩＤ５は多数ユーザのサービスを行うオンライン・トランザクションたデータベース処理に用いられる。一般的なディスクアレイは、前述のキャッシュ制御とディスクの並列動作の両方を取り入れて性能を向上させている。

【０００８】一方で、大容量で低コストのディスクアレイの要求が強くなってきている。この場合、単一のディスクアレイコントローラに多数のディスク装置を接続する構成が考えられるが、ディスクアレイ制御装置内の制御プロセッサの性能の限界のために容量は増加するが性能は伸びなくなってしまう問題が発生する。制御プロセッサはホストからの入出力要求を解釈実行するために設けられる。制御しなければならないディスク装置の台数に比例した処理性能が要求されることから、現状では数台から数１０台規模のディスクアレイがほとんどである。

【０００９】この問題を解決する方法として、制御装置を階層化する方法が特開平７－４４３２２号公報に述べられている。これは制御装置を階層化することで、各制御装置が制御しなければならない台数を削減することが目的である。例えば、一つの制御装置で５台のディスク装置を制御できるとする。最上位にこの制御装置を１つ設け、その制御装置の複数の出力をそれぞれ異なる制御装置の入力に接続する。こうすることで全体で多数台のディスク装置を制御することが可能になる。

【００１０】

【発明が解決しようとする課題】特開平７－４４３２２号公報で述べられている階層化されたディスクアレイ制御装置において、キャッシュの適用方法が課題となる。最も簡単な方法は、最上位の制御装置にのみキャッシュを持つことである。しかし、単一の制御装置に接続できるキャッシュ容量は多くないため、多数台のディスク装置に対して小容量キャッシュでは十分な性能が期待できない。そこで、階層化された制御装置の各々にキャッシュを設ける方法が考えられる。この場合、各キャッシュが前述のＬＲＵ制御を行うことになる。階層化されたキャッシュの各々で、最も最近サクセスされたデータを残す制御を行うと、キャッシュ間で重複したデータが格納されることになる。

【００１１】例えば、制御装置１に制御装置２、３が接続された階層ディスクアレイであり、各制御装置に各々キャッシュが設けられているとする。制御装置２、３には、それぞれ複数のディスク装置が接続されているとする。このとき、制御装置２に接続されたディスクへの読み出し要求が発行された時、制御装置２はディスクからデータを読み込み、ＬＲＵ制御にしたがって自キャッシュ内にデータを格納する。制御装置２はそのデータを上位である制御装置１へ転送すると、制御装置１も制御装置２と同様に自キャッシュ内にデータを格納する。この場合、上位制御装置のみ格納していればよいデータを複数のキャッシュで重複して格納する問題が発生する。なお、特開平７－４４３２２ではキャッシュの制御方法については述べられていない。

【００１２】本発明の目的は、データを複数のキャッシュで重複して格納することを抑制し、キャッシュを有効に使用する手段を提供することにある。

【００１３】

【課題を解決するための手段】上記目的を達成するために、階層化された制御装置のそれぞれに、自制御装置の階層レベルを格納するメモリを設け、データの読み出し時に前記メモリ内の前記階層レベルを読み出し、最上位制御装置でなければ自キャッシュにはデータを格納しない制御を行うようにする。また、書き込み時にはデータをキャッシュに格納するとともに、パリティ生成に必要なデータが上位キャッシュにないかサーチし、存在すれば上位制御装置から下位制御装置内のキャッシュにデータを転送しパリティ生成を行うようにする。

【００１４】

【発明の実施の形態】以下、本発明に係る階層ディスクキャッシュ制御方法を図面に示し実施例を参照してさらに詳細に説明する。

【００１５】図１は、本発明による階層ディスクキャッシュ制御方法を適用するディスクアレイ制御装置の概略構成図を示したものである。１０２は、ディスクアレイ制御装置であり、１３７、１３８はディスクアレイ制御装置１０２が入出力を行うディスク装置である。ディスクアレイ制御装置１０２は、ホストプロセッサ１０１に接続されている。ホストプロセッサ１０１からのディスク入出力要求をディスクアレイ制御装置１０２が受け取り、ディスク装置（１３７、１３８）に対して入出力を行う。ディスクアレイ制御装置１０２は、上位制御装置１０３と複数の下位制御装置（１３９、１１２）から構成されている。複数の下位制御装置（１３９、１１２）はディスク装置（１３７、１３８）の入出力を行い、上位制御装置１０３は、ホストプロセッサ１０１の入出力要求を受けつけ解釈し、複数の下位制御装置（１０４、１１２）に対して入出力要求の分配を行う。

【００１６】上位制御装置１０３は、制御プロセッサ１０６（以下ＣＰＵと略す）、キャッシュコントロールメモリ１０７（以下ＣＣＭと略す）、マイクロプログラム１０８、ホストインタフェース制御部１０４、上位キャッシュ１０５、スイッチ１０９、インタフェース制御部

（１１０、１１１）から構成されている。上位制御装置１０３の主な機能は、ホストプロセッサ１０１のディスク入出力要求を解釈し、その結果を複数の下位制御装置（１３９、１１２）に分配することである。これらの制御はＣＰＵ１０６によって行われ、この制御の手順や方法が記されたものがマイクロプログラム１０８と呼ばれるソフトウエアである。

【００１７】マイクロプログラム１０８は通常は半導体メモリに格納されており、ＣＰＵ１０６から読み出し・実行が行われる。ホストインタフェース制御部１０４は、ホストプロセッサ１０１との接続を制御し、入出力要求を受け付け、データの送受信を行う。スイッチ１０９は複数の下位制御装置（１３９、１１２）に入出力要求を分配する機能を持つ。ＣＰＵ１０６からの指示で、入出力要求をどの下位制御装置に分配するのかが決定される。インタフェース制御部（１１０、１１１）は下位制御装置（１３９、１１２）との接続を制御し、入出力要求の転送やデータの送受信を行う。上位制御装置１０３には、入出力を高速化するための上位キャッシュ１０５がある。

【００１８】上位キャッシュ１０５は、ホストプロセッサ１０１から要求のあったデータを一時的に格納しておくことでディスク装置（１３７、１３８）へのアクセスを削減し入出力処理を高速化するために設けられる。ＣＣＭ１０７は上位キャッシュの制御方法に関する情報が格納されている。

【００１９】下位制御装置（１３９、１１２）内は、上位制御装置１０３との接続を制御するインタフェース制御部（１１３、１２５）、下位制御装置（１３９、１１２）を制御するＣＰＵ（１１４、１２６）とマイクロプログラム（１１６、１２８）、ディスク装置（１３７、１３８）のデータを一時的に格納する下位キャッシュ（１２４、１３０）、下位キャッシュ（１２４、１３０）に関する制御情報が格納されたＣＣＭ（１１５、１２７）、ディスクコマンドを分配するスイッチ（１２３、１３１）、ディスク装置（１３７、１３８）との接続を制御するインタフェース制御（１１７～１２１、１３２～１３６）、パリティ制御（１２２、１２９）から構成されている。

【００２０】下位制御装置（１３９、１１２）の主な機能は、性能と信頼性を向上させるディスクアレイ制御を行うことである。ディスクアレイには、データやパリティの格納方法により複数の種類があるが、本実施例では、それらのいずれも使用可能である。それらの主な種類は次の通りである。ＲＡＩＤ１はミラーリングであり、２台のディスクに同一データを格納することでディスクの障害に対する信頼性を高めている。読み出し時には、２台のディスクの内どちらか早いほうのディスクから読むことで単一ディスクに比べ高速である。ＲＡＩＤ３は、単一データをビットあるいはバイト毎にストライピング（分割）することで、データ転送性能を向上可能である。

【００２１】ＲＡＩＤ５は入出力ブロックを単位として複数のディスクにストライピングを行うレベルである。ＲＡＩＤ３では単一入出力要求を小さな単位でストライピングするのに対し、ＲＡＩＤ５はそれよりも大きな単位でストライピングを行う。ＲＡＩＤ３は単一ユーザの入出力を高速化することが主な目的であるが、ＲＡＩＤ５は複数ユーザの入出力を並列に実行することが主な目的である。従って、ＲＡＩＤ３は特に大規模なデータ入出力が要求される画像等を処理するマルチメディアや科学技術計算に用いられる。ＲＡＩＤ５は多数ユーザのサービスを行うオンライン・トランザクションたデータベース処理に用いられる。

【００２２】ＲＡＩＤ１、３、５ではいずれの場合も、データ格納時に冗長データも同時にディスクに書き込むことで信頼性も向上している。具体的には、ＲＡＩＤ１の場合は、同一データを複数のディスクに書き込むことで冗長性を持たせている。またＲＡＩＤ３、５では、データ複数のディスクにストライピングするが、この時、旧データと旧パリティと新データとの排他的論理和を新パリティとしてディスクに書き込む。こうすることで、いずれかのディスクが障害を起こしても、ＲＡＩＤ１、３、５いずれの場合も障害を起こしたディスクのデータを回復することが可能である。

【００２３】例えば、ＲＡＩＤ３において以下のようにデータが格納されているとする。

ディスク１　　データ　＝１０１０１０１０
ディスク２　　データ　＝１１１１０００
ディスク３　　パリティ＝０１０１１０１０

この状態でディスク２が障害を起こし読み書きができなくなった場合、ディスク１とディスク３の排他的論理和を演算する。

ディスク１　　データ　＝１０１０１０１０
ディスク３　　パリティ＝０１０１１０１０
論理和　　＝１１１１０００（ディスク２のデータ）

このように、ディスク２のデータを再現することが可能である。図１のパリティ制御（１２２、１２９）はパリティの生成を行う機構である。

【００２４】本発明では図１に示すように、キャッシュが上位制御装置１０３と下位制御装置（１３９、１１２）間で階層化されている。キャッシュを階層化することの効果は、要求アクセス時間やコストに応じて、構成を柔軟に変更できることである。一例として、上位のキャッシュ１０５はコストが高いが高速な半導体メモリで構成し頻繁にアクセスされるデータを保持しておき、下位キャッシュは上位キャッシュに比べ遅いが低コストな半導体メモリで構成するといったことが考えられる。ディスク制御装置のほとんどはキャッシュが単一であった

ために、このような柔軟な構成をとることができなかった。

【００２５】本発明の特徴は、キャッシュが階層化されたディスクアレイ制御装置において、上位制御装置１０３内と複数の下位制御装置（１０４、１１２）内にキャッシュ（１０５、１２４、１３０）を設け、互いのキャッシュに格納されたデータの重複を無くす制御を行うことでキャッシュの利用効率を向上することである。

【００２６】図２および図３は、本発明における実施例の概要を示している。図２では動作概要を示し、図３では効果を示している。２０１は上位キャッシュであり、上位キャッシュ２０１には複数の下位キャッシュ（２０５、２０６）が接続されており、階層キャッシュ構造になっている。ホストプロセッサから入力要求が発生した場合の動作を示している。ディスク装置に格納されたデータ２０４の入力要求が発生した場合、下位キャッシュにデータを格納するかどうかを階層レベルにより決定する。階層レベルとは、ホストプロセッサに最も近いキャッシュを最上位（階層レベル＝０）２０８と定義し、それ以外は階層レベル＝１（２０９、２１０）である。キャッシュに格納する格納するかどうかは、自キャッシュが最上位キャッシュかどうか判定することで決定する。下位キャッシュ２０５は、最上位キャッシュではないので、下位キャッシュ２０５には読み込まれたデータは格納しない２１２。読み込まれたデータは上位キャッシュ２０１に転送される。

【００２７】ここで、下位キャッシュと同様に最上位キャッシュかどうか判定する（２１１）。ここでは最上位キャッシュであるため、キャッシュに読み込まれたデータは格納され（２０２）、そのデータは同時にホストプロセッサに転送される。この時、最上位キャッシュが一杯になった場合、自キャッシュ内のいずれかのデータの追い出し処理を行う。追い出しデータとして選択されたデータ２０３は、所定の下位キャッシュへ転送される（２０７）。下位キャッシュでは、上位から転送されたデータについては、格納するかどうかの判定は行わない。格納するかどうかの判定を行うのは、自キャッシュより上位にデータを転送する時のみであり、上位から転送されたデータは無条件に格納する。これは、上位キャッシュは、最も頻繁に使用されるデータのみを格納し、下位キャッシュには上位キャッシュに格納しきれなかったデータを格納することで、キャッシュのヒット率を向上させるためである。

【００２８】図２の制御を行うことで、図３のような効果が得られる。図３は、従来のキャッシュ制御と、本発明の実施例における階層キャッシュの効果を比較した図である。従来のキャッシュ制御は、全てのキャッシュにおいてまったく同じ制御を行っているために、データの重複が生じる。上位キャッシュ１１０１に１２個のデータが格納でき、下位キャッシュ（１１０２、１１０３）にはそれぞれ６個のデータが格納できる場合を示している。

【００２９】ホストプロセッサから入力要求が発生すると、上位キャッシュ１１０１、下位キャッシュ（１１０２、１１０３）それぞれで、最も最近アクセスされたデータを残す制御を行う。そのため、キャッシュに格納されたデータが重複するために、キャッシュを無駄に使用することになる。本発明では、上位キャッシュ１１０４には従来と同様に最も頻繁にアクセスされるデータを残し、下位キャッシュ（１１０５、１１０６）には、上位キャッシュ１１０４に転送したデータは格納しない制御を行っているため、キャッシュに格納されるデータに重複が生じない。そのため、従来は重複データが格納されていた領域は空き領域となり、この領域に新しいデータを格納することができるようになる。

【００３０】図４は、本発明における実施例のマイクロプログラム３０１の構造を示してる。上位制御装置１０３と下位制御装置（１３９、１１２）ともに構成は同じである。３０２はコマンド制御、３０３は階層キャッシュ制御、３０４はキャッシュ管理リスト、３０５はディスク読み込み＆先読み制御、３０６はディスク書き込み＆まとめ書き制御、３０７は階層キャッシュ初期設定制御である。自制御装置より上位の装置からの要求はコマンド制御３０２で受け付けられ解釈される。その結果、読み込み処理であればディスク読み込み＆先読み制御３０５が呼び出され、書き込み処理であればディスク読み込み＆先読み制御３０６が呼び出される。ディスク読み込み＆先読み制御３０５、ディスク読み込み＆先読み制御３０６内ではそれぞれ階層キャッシュ制御３０３が呼びだされ自制御装置内のキャッシュ制御が行われる。その際、キャッシュ管理リスト３０４が使用される。キャッシュ管理リスト３０４は自キャッシュの使用リスト、空きリストなどが格納されている。階層キャッシュ初期設定制御３０７は、主に装置起動時に実行され、階層キャッシュの制御に必要な情報を設定する。これらの情報は次に説明する。

【００３１】図５は、階層キャッシュの制御に必要な情報が格納されたＣＣＭ（１０７、１１５、１２７）のデータ構造を示している。これらの情報は、装置起動時に階層キャッシュ初期設定制御３０７によって設定される。４０１は階層レベルフィールドであり、ホストプロセッサに最も近いキャッシュが階層レベル０であり、それ以外は階層レベル１である。キャッシュ制御モード４０２は、キャッシュの先読みやまとめ書き制御を行うかどうかのフラグが設定されている。キャッシュ使用上限値４０３とキャッシュ使用下限値４０４は、キャッシュのあふれや先読み制御をどの時点で行うかの数値が格納されている。キャッシュの使用状況がキャッシュ使用上限値に達したら、下位制御装置またはディスク装置へデータを転送し、未使用領域を増やす。また、キャッシュ

使用下限値を下回ったら下位制御装置またはディスク装置からデータの先読みを行いキャッシュに格納されているデータを増やす。キャッシュ使用リストポインタ４０５とキャッシュ未使用リストポインタは図４のキャッシュ管理リスト３０４へのポインタを示しており、キャッシュの確保や開放を行う際に使用される。

【００３２】図６は、図４のマイクロプログラム３０１内のコマンド制御３０２のフローを示している。ステップ５０１は、上位装置から転送されたディスク入出力コマンド受付処理である。ステップ５０２はコマンドの判定を行う。ここでは、コマンドが入力要求なのか出力要求なのかが判定され、入力要求（ＲＥＡＤ）であればステップ５０３に進み、出力処理（ＷＲＩＴＥ）であればステップ５０６に進む。

【００３３】ステップ５０３は、ディスク読み込み＆先読み制御を行う。この処理は後で詳細に説明する。ステップ５０４は、ステップ５０３で読み込まれたデータに対してキャッシュ制御を行う。ステップ５０５は上位装置へ終了報告を行う。ステップ５０６は、上位装置から転送されたデータの階層キャッシュ制御を行う。ステップ５０７はディスク書き込み＆まとめ書き制御を行う。この処理は後で詳細に説明する。ステップ５０８は、上位装置へ終了報告を行う。

【００３４】図７は、図４の階層キャッシュ制御３０３のフローを示している。ステップ６０１はコマンド判定処理であり、出力要求か入力要求かを判定する。入力要求であればステップ６０２に進み出力要求であればステップ６０９に進む。ステップ６０２では、ＣＣＭ（図１の１０７、１１５、および、１２７）内の階層レベルフィールド（図５の４０１）の読み出しを行う。このフィールドには、最上位キャッシュであれば０が、それ以外のキャッシュであれば１が格納されている。ステップ６０３は自キャッシュが最上位キャッシュかそうでないかの判定を行う。最上位キャッシュであればステップ６０４に進み、そうでなければステップ６０６に進む。ステップ６０４では、ＬＲＵ処理を行う。ＬＲＵ処理とは、キャッシュがフルになった時にどのデータを追い出すかを決定するアルゴリズムであり、キャッシュの管理に一般に用いられている。

【００３５】ステップ６０４では、読み込み要求に対して転送されたデータを最優先にキャッシュに格納するため、もしその時キャッシュがフルになっていた場合、いずれかのデータをキャッシュから追い出す処理を行わなう。ステップ６０５では、上位装置へデータを転送する。

【００３６】ステップ６０６、６０７は、自キャッシュが最上位キャッシュではなかった時の処理である。最上位キャッシュでなかった場合は、キャッシングしない制御を行う。これは、さらに上位にキャッシュが存在するため上位のキャッシュに格納されていれば自キャッシュにデータを格納する必要がないためである。ステップ６０６では、上位装置にデータを転送する。ステップ６０７ではキャッシュ領域を解放する。

【００３７】ステップ６０４、６０５、６０６、６０７により、自キャッシュが最上位キャッシュであればキャッシュに格納し、そうでなければキャッシュには格納しない処理を実現してる。この判定のためにＣＣＭ内に階層レベルを示す情報がキャッシュ毎に格納されている。

【００３８】ステップ６０９は書き込み処理の場合に実行され、通常のＬＲＵ処理が行われる。この階層キャッシュ制御により、階層化されたキャッシュ構成において、各キャッシュ間に重複するデータをなくすことができ、キャッシュの利用効率を向上することが可能になる。

【００３９】図８は、キャッシュ管理リスト３０４のデータ構造を示している。４０５は図５におけるキャッシュ使用リストポインタであり、４０６はキャッシュ未使用リストポインタである。キャッシュ使用リストポインタ４０５には、自キャッシュ内で使用されている領域がリスト形式で格納されており、各リスト内は次のリストへのポインタ７０３、使用領域識別子７０４が格納されている。リストの最後の次のリストへのポインタ７０３はリストの最後を示す値が格納される。このリストは、参照されるとリストの最も最後にリストチェインを切り替える制御が行われる。

【００４０】これにより、最も最近アクセスされた領域はリストの最も最後におかれ、最も長くアクセスされなかった領域はキャッシュ使用リストポインタ４０５に最も近いリストにチェインされている。そのため、キャッシュがフルになり、いずれかの領域を解放しなくてはならなくなったときにはキャッシュ使用リストポインタ４０５に最も近いリストが選択することでＬＲＵ制御を実現している。キャッシュ未使用リストポインタ４０６に示されるリストは、使用されていないキャッシュ領域がチェインされており、キャッシュを確保する場合にはリストの先頭か領域が選択される。

【００４１】図９は、ディスク読み込み＆先読み制御のフローを示している。ステップ８０１では上位装置より要求されたデータがキャッシュに格納されているかどうかの判定を行う。これは図８のキャッシュ使用リストをサーチすることで実現できる。ステップ８０２でもしキャッシュにデータが格納されていたら（キャッシュヒット）処理を終了させる。キャッシュに格納されていなければ（キャッシュミスヒット）ステップ８０３に進む。ステップ８０３では、キャッシュ領域を確保する。これは、図８のキャッシュ未使用リストポインタが示しているリストを選択することで実現できる。ステップ８０４ではディスク入力要求を生成し、ディスク装置からデータの入力を行う。ステップ８０５は先読み制御であるが、この処理はバックグラウンドで実行される。先読み

処理の中断は、図５キャッシュ使用上限値４０３に達したとき、または次の入力コマンドが発生したときである。

【００４２】図１０は、ディスク書き込み＆まとめ書き制御のフローを示している。ステップ９０１では、上位装置から転送されたデータを格納するためのキャッシュ領域を確保する。これはキャッシュ未使用リストポインタ４０６に示されるリストを選択することで実現できる。ステップ９０２では、自キャッシュ内のディスクへ書き込んでいないデータ量がＮを超えたかどうかを判定している。もし、Ｎを超えていたらディスクへの書き込みを実行する。これは、小データの場合に、要求が発生する毎にディスクへの書き込みを行っていると処理時間が増加するため、ある程度まとまった単位でディスクへの書き込みを行うことで性能を向上させるための処理である。ステップ９０３、９０４はパリティ生成を行うための処理である。

【００４３】前述の通り、ディスクアレイではパリティを生成しディスクに格納することで信頼性を向上させている。パリティの生成には、旧データ、旧パリティ、新データが必要であり、ステップ９０３ではこれらのデータが上位キャッシュにないかサーチする。これは、本発明はデータを重複しないようにキャッシュ制御を行っているため、最新データが上位装置に存在する可能性があるためである。もし、上位キャッシュ、自キャッシュにパリティ生成に必要なデータが存在しない場合は、下位装置へアクセスしデータを集める。ステップ９０４では旧データ、旧パリティ、新データの排他的論理和をとり、パリティを生成する。ステップ９０５では、書き込みデータ（新データ）とパリティを所定のディスクへ書き込むことで処理が終了する。

【００４４】図１１では、図４における階層キャッシュ初期設定制御３０７によって、図１におけるＣＣＭ（１０７、１１５、１２７）内の階層レベルフィールド（図５の４０１）にどの様な値が格納されるかを示している。これまでの実施例では２階層のキャッシュで説明してきたが、図１１に示すように、３階層、あるいはそれ以上でも本発明の効果が得られる。最上位制御装置１００１に接続された中間制御装置（１００２、１００３）、さらに中間制御装置（１００２、１００３）に接続された最下位制御装置（１００４、１００５）のような構成の場合、階層レベルフィールドは次のようになる。

【００４５】最上位制御装置１００１は０、中間制御装置（１００２、１００３）は１、最下位制御装置（１００４、１００５）は１に設定することで、各階層間でデータが重複することは無くなる。これまでは装置が正常に動作している場合を述べてきたが、ダイナミックに階層レベルを変更したほうが良い場合がある。装置あるいはキャッシュに障害が発生した場合である。最上位制御装置１００１内のキャッシュが障害を起こした場合は、その次の階層のキャッシュ（図１１では中間制御装置１００２、１００３）が最上位キャッシュ（階層レベルフィールド＝０）に設定される。これにより、障害時においても本発明の効果を十分に得ることができるようになる。

【００４６】

【発明の効果】本発明では、階層化されたキャッシュで構成されるディスクアレイにおいて、下位キャッシュでは、上位にデータを転送する時は自キャッシュにデータを格納しない制御を行う。これにより上位キャッシュは、最も頻繁に使用されるデータのみを格納され、下位キャッシュには上位キャッシュに格納しきれなかったデータを格納されることになるため、キャッシュのヒット率を向上させるためである。

【図面の簡単な説明】

【図１】本発明における実施例の全体図を示す図である。

【図２】本発明の実施例における動作概要を示す図である。

【図３】本発明の実施例における効果を示す図である。

【図４】本発明の実施例におけるマイクロプログラム構成を示す図である。

【図５】本発明の実施例におけるキャッシュ制御メモリ（ＣＣＭ）構造を示す図である。

【図６】本発明の実施例におけるコマンド制御フローを示す図である。

【図７】本発明の実施例における階層キャッシュ制御フローを示す図である。

【図８】本発明の実施例におけるキャッシュ管理リスト構造を示す図である。

【図９】本発明の実施例におけるディスク読み込み＆先読み制御フローを示す図である。

【図１０】本発明の実施例におけるディスク書き込み＆制御フローを示す図である。

【図１１】本発明の実施例における階層レベルを示す図である。

【符号の説明】

１０１　ホストプロセッサ、
１０２　ディスクアレイ制御装置、
１３７、１３８　ディスク装置、
１０３　上位制御装置、
１３９、１１２　下位制御装置、
１０５　上位キャッシュ、
１２４、１３０　下位キャッシュ、
１０７、１１５、１２７　キャッシュ制御メモリ。

【図１】

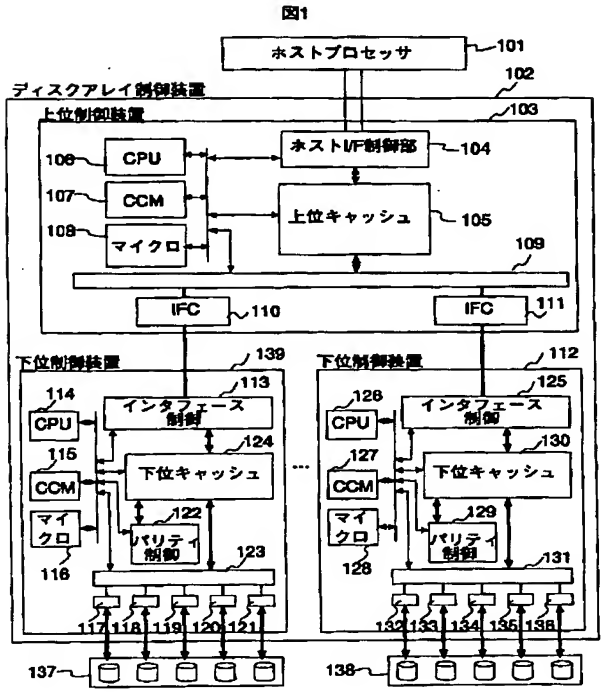


図1
ホストプロセッサ
101
ディスクアレイ制御装置
102
上位制御装置
103
CPU
106
CCM
107
マイクロ
108
ホストI/F制御部
104
上位キャッシュ
105
109
IFC
110
IFC
111
下位制御装置
139
112
インタフェース制御
113
125
CPU
114
CCM
115
マイクロ
116
下位キャッシュ
124
130
パリティ制御
122
129
123
131
127
128
117 118 119 120 121
132 133 134 135 136
137
138

【図２】

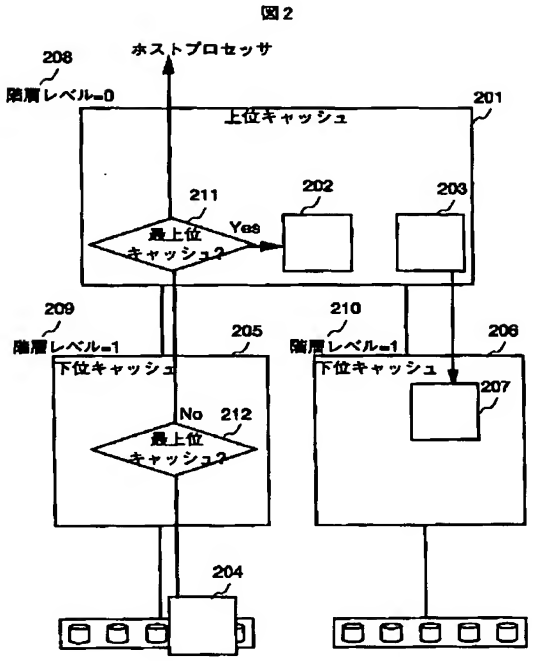


図2
ホストプロセッサ
208
階層レベル=0
201
上位キャッシュ
211
202
203
最上位キャッシュ?
Yes
209
階層レベル=1
205
210
階層レベル=1
206
下位キャッシュ
下位キャッシュ
207
No
212
最上位キャッシュ?
204

【図３】

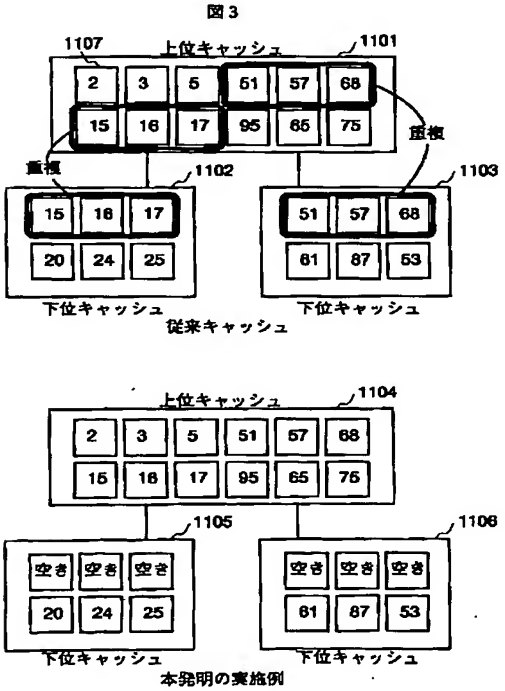


図3
1107
上位キャッシュ
1101
2 3 5 51 57 68
15 16 17 95 65 75
重複
重複
1102
1103
15 16 17
20 24 25
51 57 68
61 87 53
下位キャッシュ
下位キャッシュ
従来キャッシュ
上位キャッシュ
1104
2 3 5 51 57 68
15 16 17 95 65 75
1105
1106
空き 空き 空き
20 24 25
空き 空き 空き
61 87 53
下位キャッシュ
下位キャッシュ
本発明の実施例

【図４】

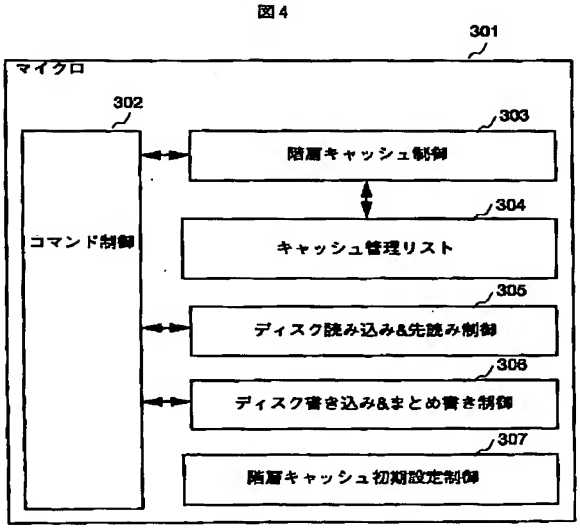
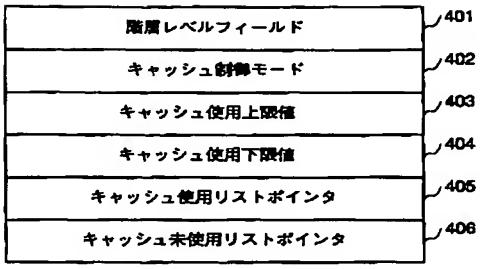


図4
301
マイクロ
302
コマンド制御
303
階層キャッシュ制御
304
キャッシュ管理リスト
305
ディスク読み込み&先読み制御
306
ディスク書き込み&まとめ書き制御
307
階層キャッシュ初期設定制御

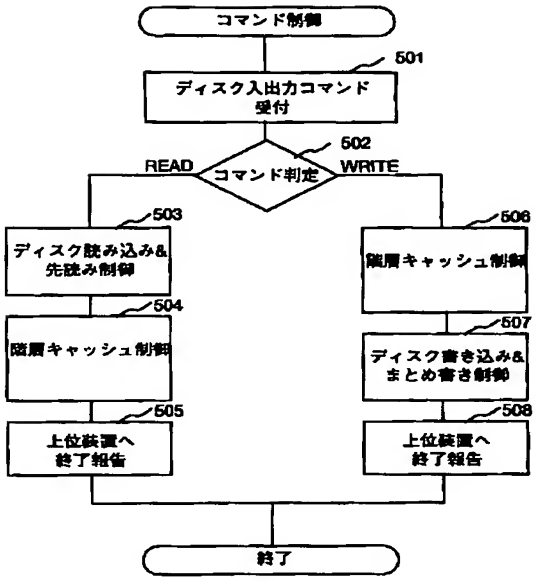
【図５】

図５



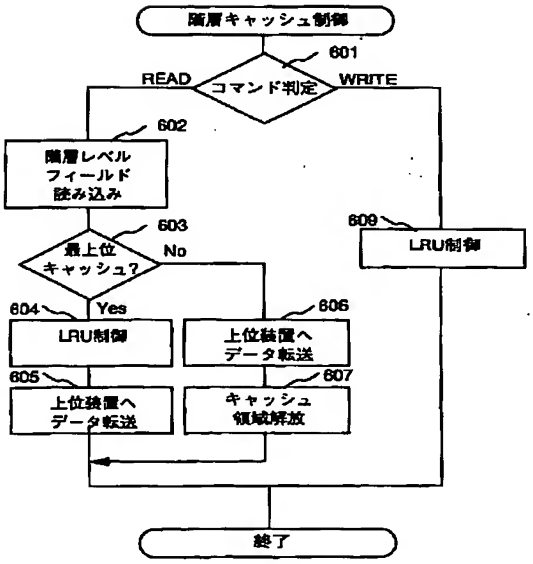
【図６】

図６



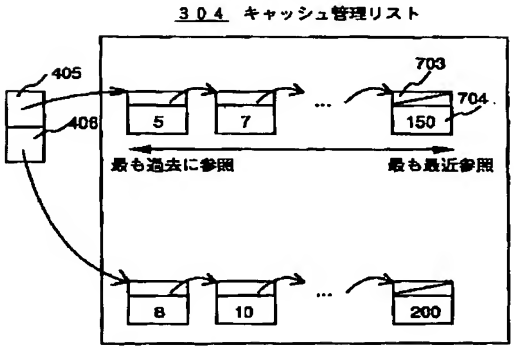
【図７】

図７



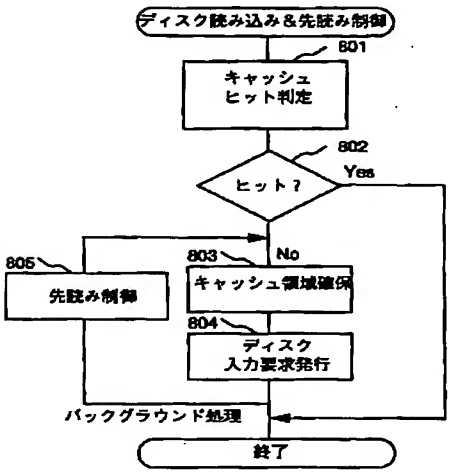
【図８】

図８



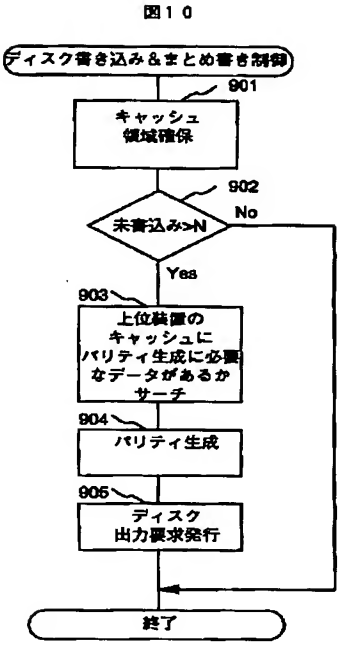
【図９】

図９



【図１０】

図１０



【図１１】

図１１